

Kipas Angin Pengikut Manusia Berdasarkan Wajah

Wahyudi Prasatia^{1*}, Kamarudin^{2*}

* Batam Polytechnics

Electrical Engineering study Program

Parkway Street, Batam Centre, Batam 29461, Indonesia

E-mail: kamarudin@polibatam.ac.id

Abstrak

Teknologi bisa membuat aktifitas keseharian kita lebih mudah. Salah satu contoh kipas angin tracking manusia secara otomatis. sistem kipas angin yang dirancang dapat mengenali wajah manusia, wajah manusia digunakan sebagai set point posisi kipas angin. Kipas angin dapat membedakan wajah manusia dengan benda lain yang ada disekitarnya. Manusia tidak perlu lagi mengarahkan kipas angin secara manual. Wajah manusia dideteksi menggunakan kamera *smartphone android*. Wajah dideteksi menggunakan metode *Haar Cascade*. Metode *Haar Cascade* telah tersedia di library *openCV*. Tracking sistem menggunakan motor DC, Motor DC dikontrol dengan metode PID kontrol. Metode tuning kontrol PID menggunakan *Zigler Nichols*. Sistem komunikasi bluetooth digunakan untuk komunikasi antara *smartphone android* dengan sistem tracking. Hasil percobaan tracking wajah manusia terdapat error 4.41%, dengan tingkat keberhasilan 95,59%. hasil pengujian deteksi wajah tingkat keberhasilan 100%, dengan posisi wajah dalam keadaan frontal. Dari hasil percobaan akurasi kipas angin dapat mendeteksi wajah sebesar 95%.

Kata kunci: *Haar Cascade*, PID, *OpenCV*, *Zigler Nichols*

Abstract

Technology can make daily activities our easier. One example of a fan track down the human automatically. The fan system designed to recognize a human face, the human face is used as a set point of the fan position. The fans can distinguish human faces from other objects around them. The humans do not need to direct the fan manually. The human faces are detected using an *Android smartphone camera*. The face was detected using the *Haar Cascade method*. The *Haar Cascade method* is available in the *openCV library*. Tracking system uses *DC motor*, *dc motor* is controlled by the *PID control method*. The tuning method of *PID control* uses *Ziegler Nichols*. The *bluetooth communication system* is used for communication between *Android smartphones* with tracking systems. The results of human face tracking experiments have a 4.41% error, with a success rate of 95.59%. The results of face detection testing 100% success, with the face position in a frontal state. From the results of the experiment, the accuracy of the fan can detect faces of 95%.

Keywords: *Haar Cascade*, PID, *OpenCV*, *Zigler Nichols*

1. Pendahuluan

Teknologi yang semakin dinamis menuntut manusia untuk lebih inovatif. Banyaknya cara dalam memecahkan masalah teknologi pada masyarakat, membuat sistem yang ada juga semakin membaik. Hal ini dapat membantu manusia berkerja lebih praktis, termasuk dalam memanjakan kehidupan

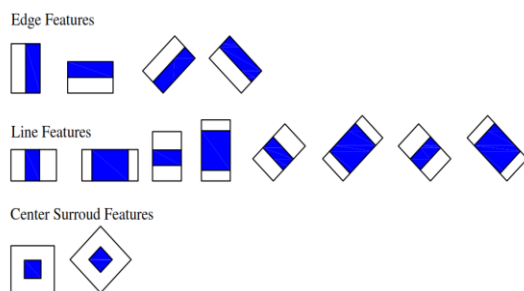
manusia. Dalam hal ini peneliti akan membahas tentang bagaimana kipas dapat autotracking terhadap manusia. Sekarang ini telah banyak cara dan metode yang muncul untuk membuat kipas *tracking* manusia secara otomatis. Pada umumnya kipas angin, untuk mendapatkan jangkauan angin yang luas, kita butuh mengatur kipas ke *mode swing*, lalu seseorang terpaksa harus mengarahkan kipas ke arahnya

secara manual, untuk mendapatkan angin yang diinginkan, tentu ini tidak praktis.

2. Tinjauan Pustaka

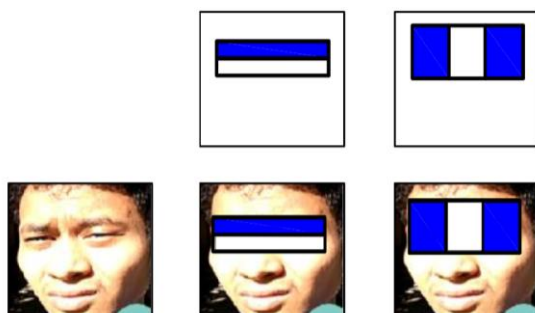
2.1 Sistem Deteksi Wajah

Deteksi wajah menggunakan kamera membutuhkan sebuah metode, dalam penelitian ini metode yang akan digunakan yaitu *Haar Cascade Classifier* yang telah tersedia pada *OpenCV library*, pada penelitian ini *OpenCV library* akan dijalankan di atas *platform android*. *OpenCV* merupakan *library* terbuka untuk komputer *vision*. *OpenCV* dirancang untuk menyederhanakan komputasi yang berguna pada aplikasi *realtime* yang memiliki 500 lebih fungsi yang dapat digunakan dalam mempercepat proses pengolahan citra [1]. Pada awalnya penelitian tentang *Haar Cascade Classifier* dilakukan oleh Viola ddk (2001)[2]. Sebuah citra wajah dibagi atas beberapa fitur yang disebut dengan *Haar feature* seperti Gambar 1.



Gambar 1 Haar-like Feature.

Haar feature ini akan dilakukan penghitungan dengan citra wajah seperti Gambar 2.



Gambar 2 Haar Feature Pada Citra Wajah.

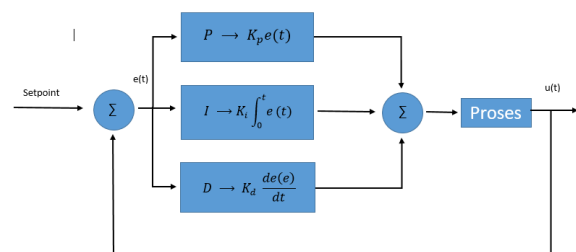
Haar feature pada citra wajah digunakan untuk menghitung perbedaan intensitas warna yang ada pada area-area wajah, untuk mencari nilai *threshold* dilakukan dengan mengurangi area putih dengan area hitam[4][5][6][7].

$$\Delta = \frac{1}{n} \sum_{hitam}^n I(x) - \frac{1}{n} \sum_{putih}^n I(x) \quad (1)$$

2.2 Kendalai PID

Kendali PID banyak digunakan untuk sistem dinamik, dengan PID kontrol sistem dinamik yang awalnya tak terkendali dapat dikendalikan, diperlukan kestabilan dalam menggunakan kendali PID membuat sistem sesuai dengan yang diinginkan, dengan adanya kendali *Propotional* dan *Derivative* telah mampu meningkatkan kinerja sistem mengatasi *error steady state* atau menjaga kestabilan sistem dinamik[3][4][5].

Rancangan kendali PID dapat digambarkan dalam bentuk diagram blok pada gambar 3



Gambar 3 Diagram Blok PID

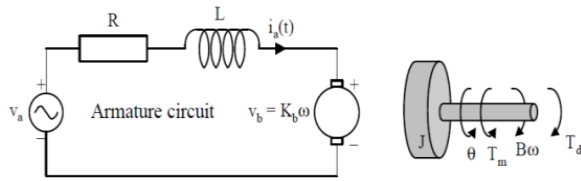
Sinyal kontrol yang digunakan dalam proses PID didapat dari *error*, adanya konstanta K_p , K_i , K_d , merupakan *tunning* parameter dari PID untuk mengendalikan *plan* atau proses. PID dapat disusun dalam persamaan berikut :

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (2)$$

Persamaan PID ini akan di implementasikan menjadi sebuah algoritma yang dapat diterapkan di dalam mikrokontroler sehingga mampu mengontrol posisi motor DC. Nilai pada sensor *rotary encoder* menjadi referensi pengendali PID untuk mengetahui *feedback error* dari sistem.

2.3 DC Motor

Motor adalah mesin yang membutuhkan arus searah, merubah kumparan medan menjadi gerak mekanik, motor DC sangat *flexible* karena mudah menyesuainya dengan berbagai pengendali kecepatan dan posisi[6][7][8][9][10][11,12]. Adapun rangkaian elektrik dan model mekanik dari DC motor seperti gambar 4.



Gambar 4 Model Mekanik dan elektrik DC Motor[6].

Dimana R = armature resistance, L = armature inductance, J = moment of inertia, K_t = motor torque constant, K_e = back emf constant, B = viscous friction dan T_m = motor torque

Untuk itu maka didapatkan persamaan matematika :

$$J \frac{d\omega(t)}{dt} + B\omega(t) = T_m(t) \quad (3)$$

$$L \frac{di_a(t)}{dt} + Ri_a(t) = v_a(t) - v_b(t) \quad (4)$$

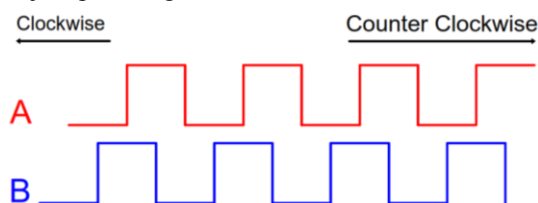
$$v_b = K_e \omega(t) \quad (5)$$

$$T_m(t) = K_t i_a(t) \quad (6)$$

Pada penelitian ini Motor DC akan dikendalikan dengan bantuan *feedback* sensor agar dapat di atur sudut dan posisinya, yang digunakan dalam mengatur arah kipas.

2.4 Rotary Encoder

Rotary Encoder akan digunakan untuk mendeteksi posisi dari obyek yang diberi rentang 120 derajat pada penelitian ini. *Rotary Encoder* adalah perangkat yang digunakan untuk mengkonversi pergerakan putaran menjadi pulsa digital.



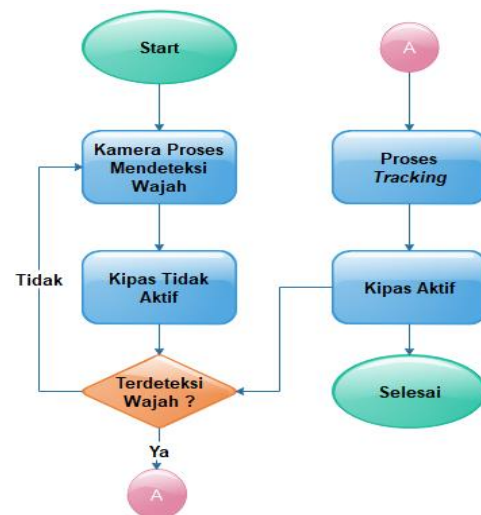
Gambar 5 Ilustrasi Output Rotary Encoder

Rotary Encoder diputar searah jarum jam maka Output sama dengan phase A pada Gambar 5, sedangkan jika diputar berlawanan arah jarum jam maka pulsa yang keluar phase B, dimana nilai pulsa cepat 90 derajat dari fasa A[15]. Dengan menggunakan bantuan mikrokontroler pulsa yang dikeluarkan oleh *encoder* dapat diolah menjadi data *Count*.

3. Metode Penelitian

3.1 Diagram Alur Proses Pencarian Fitur Wajah

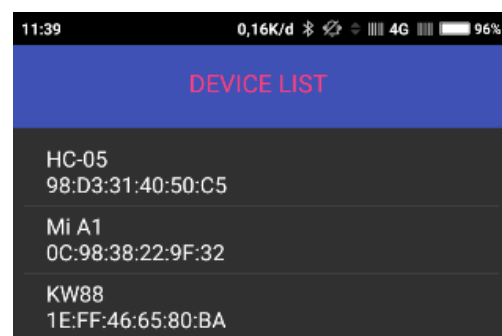
Pada Gambar diagram 6, proses yang dilakukan pertama kali yang dilakukan yaitu kamera menggunakan bantuan library OpenCV melakukan proses pencarian fitur wajah, ketika objek wajah belum ditemukan maka kipas tidak diaktifkan. Setelah kamera menemukan objek wajah maka akan dilanjutkan ke proses tracking. Selama proses tracking berlangsung kipas akan diaktifkan, hal ini akan berulang terus menerus sampai kamera tidak mendeteksi lagi adanya fitur wajah.



Gambar 6 Diagram Alur

3.2 Perancangan Aplikasi Pendeteksi Wajah

Aplikasi pendeteksi wajah pada penelitian ini akan dibuat menggunakan *IDE Android Studio*, aplikasi akan berjalan untuk perangkat *android*. Aplikasi atau *software android* ini menggunakan bantuan library *OpenCv* untuk melakukan proses deteksi wajah, aplikasi disusun dengan bahasa pemrograman *java*.



Gambar 7 Daftar Perangkat Bluetooth

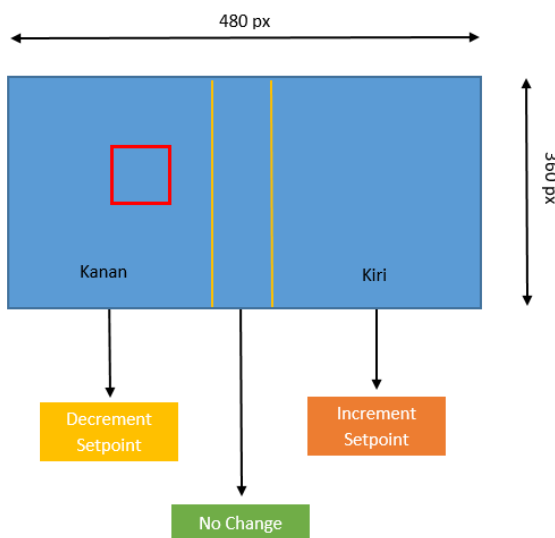


Gambar 8. *Layout Kamera*

Pada Gambar 7 merupakan *layout* yang berfungsi untuk menampung perangkat Bluetooth yang tersedia atau terkoneksi dengan perangkat *android*, juga terdapat tombol *refresh* yang berguna untuk mencari kembali perangkat yang terkoneksi, untuk penelitian ini *bluetooth* HC-05 akan dikoneksikan dengan perangkat *android* dimana komunikasi kipas angin dengan sistem *android* menggunakan *bluetooth*. Sedangkan pada Gambar 8 merupakan *layout* Kamera, tampilan ketika mengakses kamera dan proses deteksi wajah akan ditampilkan pada *layout* ini, terdapat juga tombol dengan tanda silang(x) yang berfungsi untuk menghentikan proses kamera dan deteksi wajah, kemudian sistem akan kembali ke daftar perangkat *bluetooth*.

3.3 Diagram Sistem Komunikasi

Pada bagian ini merupakan cara komunikasi data yang didapat oleh Kamera *Android Smartphone* dengan sistem pengendali arah kipas, yang dapat diamati pada Gambar 9.



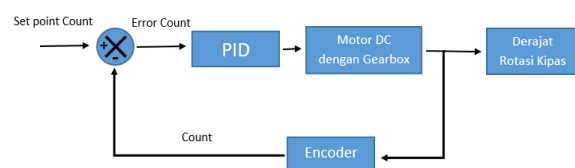
Gambar 9. Cara Berkomunikasi

Pada Gambar 9. mengilustrasikan sebuah *frame* kamera dengan ukuran 480×360 *pixel*, didalamnya

terdapat kontak berwarna merah yang menandakan sebuah objek, ketika objek berada pada area sebelah kanan maka sistem akan mengirim perintah *Increment Set point*, sebaliknya jika objek di sebelah kiri sistem akan mengirim *Increment Set point*. Perintah ini akan dikirim dari Aplikasi Pendeteksi Wajah pada *smartphone android* ke sistem pengendali melalui media *bluetooth*.

3.4 Diagram Sistem Pengendali (*Tracking*)

Pada diagram pengendali ini adalah proses yang akan terjadi pada motor DC dengan skema pada Gambar 10.



Gambar 10 Diagram Kendali

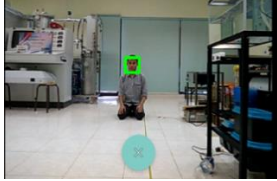
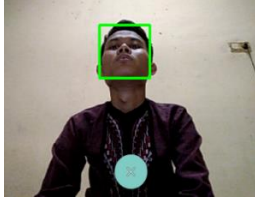


Dari Gambar 10 dapat diperhatikan bahwa nilai *Set point* akan ditentukan setelah melewati proses komunikasi, dari nilai *Set point* maka akan terbentuk nilai *error* yang diakibatkan oleh tidak sesuainya nilai pembacaan *encoder* pada motor DC, untuk menyelesaikan kasus ini maka digunakan pengendali PID yang nantinya akan mempengaruhi Motor DC untuk berputar sehingga terjadi perubahan derajat putaran pada kipas. Sedangkan *Count* merupakan penjumlahan atau pengurangan sebuah nilai berdasarkan pulsa yang dihasilkan oleh *encoder*. Proses ini akan dilakukan dengan bantuan *microcontroller Arduino Uno*.

4. Hasil Dan Pembahasan

4.1 Hasil Sistem Pendeteksian Wajah

Maka dari hasil pengujian sistem pendeteksi wajah, ditampilkan tabel hasil pengujian yang diperoleh secara ringkas pada Tabel 1.

Tabel 1 Tabel Hasil Pengujian Sistem Pendeteksi Wajah

Jenis Pengujian	Keterangan Hasil
Jarak Jangkauan Kamera	 <p>Kamera berhasil mendeteksi wajah dengan jarak $\pm 15\text{cm}$ sampai dengan $\pm 330\text{cm}$</p>
Berdasarkan Derajat kemiringan	 <p>Ketika wajah kemiringan 0° atau <i>frontal</i> maka sistem berhasil mendeteksi 100%, dan maksimum kemiringan wajah menghadap ke kiri atau ke kanan adalah $\pm 37^\circ$, kemiringan wajah hadap ke bawah $\pm 21^\circ$, kemiringan wajah hadap ke atas $\pm 40^\circ$</p>
Wajah terhalang objek	 <p>Wajah tidak terdeteksi jika objek penghalang menghilangkan kontur wajah.</p>
Berdasarkan Pencahayaan	 <p>Rata-rata histogram pencahayaan sistem dapat mendeteksi wajah antara 10.1147 sampai dengan 163.9997.</p>

4.2 Hasil Sistem Pengendali Arah Kipas

Setelah didapatkan nilai-nilai konstanta K_p , K_i , K_d maka dilakukan percobaan dengan menggunakan pengendali PID. Nilai-nilai dari respon pengendali PID tersebut diperlihatkan sebagaimana pada Tabel 2.

Tabel 2 Respon Sistem Pengendali Arah Kipas dengan Metoda

Zigler Nichols

Set Point (y_{final})	Rise Time (t_r)	Settling Time (t_s)	Settling Max (S_{max})
100 Count	74 ms	89 ms	103
Settling Min (S_{min})	Peak	% Overshoot (M_p)	
99	103 Count	3%	

Dari nilai-nilai yang ditampilkan pada Table 2 untuk nilai persentase *Overshoot* dan *Peak* didapat berdasarkan perhitungan di bawah ini:

$$Peak = S_{max} = 103 \text{ Count}$$

$$\%Overshoot (M_p) = \left| \frac{y_{final} - Peak}{y_{final}} \right| \times 100$$

$$= \left| \frac{100 - 103}{100} \right| \times 100$$

$$\%Overshoot(M_p) = 3 \%$$

4.4 Hasil Pengujian Tracking Arah Kipas Berdasarkan Wajah

Berdasarkan data pengujian maka didapat persentase rata-rata *error* dari sistem saat proses *tracking* arah kipas berdasarkan wajah yang diperlihatkan sebagaimana pada Tabel 3.

Tabel 3 Rata-Rata Persentase Error Pengujian

No.	Percobaan	Rata-rata Error (%)
1	Jarak 15 cm	16,48
2	Jarak 30 cm	14,99
3	Jarak 45 cm	12,94
4	Jarak 60 cm	9,91
5	Jarak 75 cm	4,06
6	Jarak 90 cm	1,80
7	Jarak 115 cm	1,01
8	Jarak 130 cm	1,06
9	Jarak 145 cm	1,80
10	Jarak 160 cm	2,26
11	Jarak 175 cm	2,16
12	Jarak 190 cm	2,34
13	Jarak 215 cm	1,75
14	Jarak 230 cm	1,20
15	Jarak 245 cm	1,63

16	Jarak 260 cm	1,05
17	Jarak 275 cm	1,61
18	Jarak 290 cm	1,54
19	Jarak 315 cm	1,69
20	Jarak 330 cm	1,44
Rata - rata		4,14

Berdasarkan Tabel 3 percobaan pertama sampai percobaan ke-5, sistem terdapat *error* yang sangat besar. Sedangkan percobaan ke-6 sampai dengan percobaan ke-20 sistem memiliki *error* relatif lebih kecil.

5. Penutup

5.1 Kesimpulan

Sistem kipas pengikut manusia berdasarkan wajah mampu membedakan antara manusia dengan hambatan lain berdasarkan wajah. Dengan menggunakan *library OpenCV* wajah dapat terdeteksi dengan ketentuan terdapatnya kontur wajah yang lengkap berserta objek yang menyerupai kontur wajah. Sistem Kendali Motor DC dengan menggunakan metode *tuning Zigler Nichols* membuat sistem *tracking* arah kipas berdasarkan wajah menjadi terkendali dengan baik. Kipas berhasil men-*tracking* manusia berdasarkan wajah dengan tingkat rata-rata keberhasilan 95,86 %, dan terdapat *error* 4.14%.

Referensi

- [1] Bradski, Gary R. Learning OpenCV: computer vision with the OpenCV library. Farnham ; Cambridge: O'Reilly, 2008.
- [2] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, 2001, pp. I-511-I-518 vol.1.
- [3] V. Suhas kumbhar, P. Suhas Kumbhar, Vinod B. Kumbhar, Prashant P. Zirmite, Design and Implementation of an Electronic System for Identification of Rebars in Reinforced Concrete, Vol. 03, Issue 12, 2016.
- [4] Kamarudin dan Endra Pitowarno, "Kontrol 3D Inersia Berbasis Proportional Derivative Active Force Control (PDAFC)", "The 3rd Indonesian Symposium on Robot Soccer Competition 2015", June 2015.
- [5] Kamarudin and E. Pitowarno, "2D independent actuator control based on a Proportional Derivative Active Force Control," 2015 International Electronics Symposium (IES), Surabaya, 2015, pp. 79-84.
- [6] Ankit Rastogi and Pratibha Tiwari,"Optimal Tuning of Fractional Order PID Controller for DC Motor Speed Control Using Particle Swarm Optimization," International Journal of Soft Computing and Engineering, Volume-3, Issue-2, May 2013.
- [7] Dingy'u Xue, YangQuan Chen and Chunna Zhao, "Fractional Order PID Control of A DC-Motor with Elastic Shaft:A Case Study,"Proceedings of the 2006 American Control Conference Minneapolis, Minnesota, USA, June 14-16, 2006
- [8] Saurabh Dubey, Dr. S.K. Srivastava, "A PID Controlled Real Time Analysis of DC Motor,International Journal of Innovative Research in Computer and Communication Engineering, Vol.1,Issue 8, October 2013.
- [9] Dr. Jamal A. Mohammed, " Modeling, Analysis and Speed Control Design Methods of a DC Motor" ,Eng. & Tech.Journal ,vol.29, no.1, 2011.
- [10] Dr.Ch.Chengaiiah, K.Venkateswarlu, "Comparative Study On Dc Motor Speed Control Using Various Controllers", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering,Vol.3, Issue 1, January 2014.
- [11] Vivek Shrivastva, Rameshwar Singh, "Performance Analysis of Speed Control of Direct Current (DC) Motor using Traditional Tuning Controller", International Journal of Emerging Technology and Advanced Engineering, Vol 4, Issue 5, May 2014.
- [12] Kamarudin, Endra Pitowarno dan R. Sanggar Dewanto,"3D Independent Actuators Control Based On A ProportionalDerivative Active Force Control", Vol.32, June 2016.